Synergic Cross-Layer BPaaS Monitoring & Adaptation Framework

K. Kritikos[1], C. Zeginis[1], F. Griesinger[2], D. Seybold[2] , J. Domaschka[2]

1: ICS-FORTH, Greece

2: University of Ulm, Germany

FORTH Institute of Computer Science

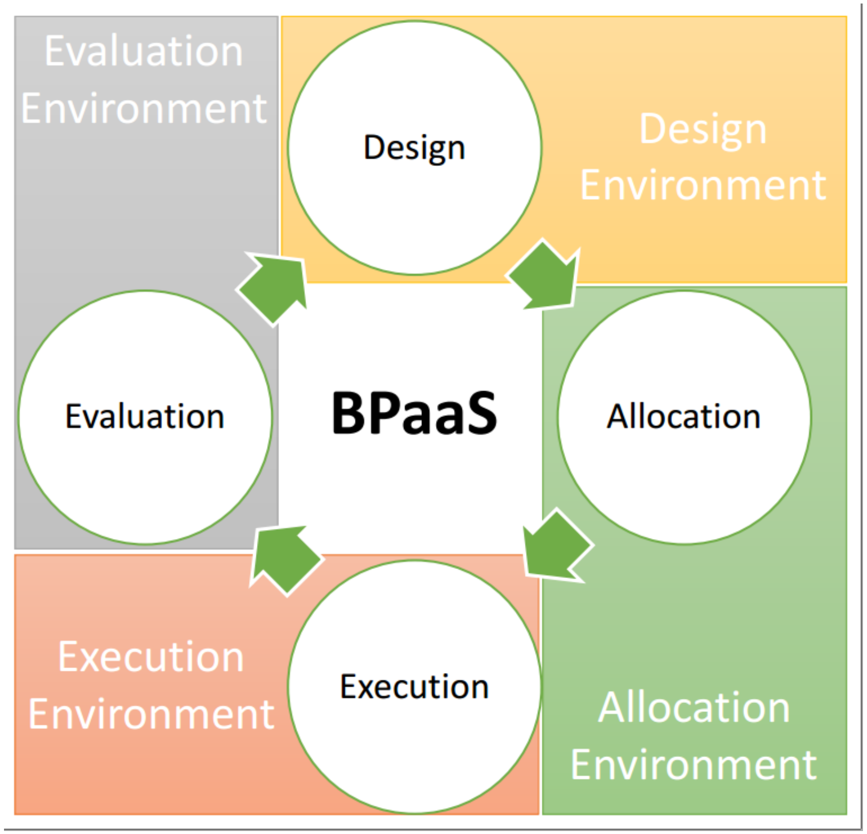
CloudSocket

Outline

- Problematic
- Solution Overview & Architecture
- CAMEL Overview & Extensions
- Validation
- Future Work Directions

Problematic – Issues

- Flexibility & cost reduction in BPs via the cloud
  - Business process as a Service (BPaaS)
- Need to handle whole lifecycle of BPaaS



Problematic – Issues

- Focusing on Execution activity, there exists the need to:
  - Monitor & Adapt BPaaS in a cross-layer manner to sustain a certain service level
- Issues
  - Many layers involved: IaaS, PaaS, SaaS, BPaaS
  - Need for flexibility in metric specification & computation
  - Need to realise layer-specific adaptation mechanisms
  - Need to coordinate such mechanisms to deal with complex, problematic situations

Problematic – Related Work Analysis on Service Adaptation

Work	Cross-Layer	Levels	Type	Dynamic	History
[7]	N	S	R	N	N
[8]	N	S	R	N	N
[9]	N	S	R	N	N
[10]	N*	ISW	R	Y*	N
[11]	Y	ISW	R	Y	N
[12]	Y	ISW	R	Y	N
[5]	Y	ISW	A	N	N
Amazon EC2	N	I	R	N	N
PaaSage [6]	N	IP	R	N	N*
[15]	Y	IS	R	N	N
Our Framework	Y	IPSW	A	Y	Y

Our
Solution

CloudSocket

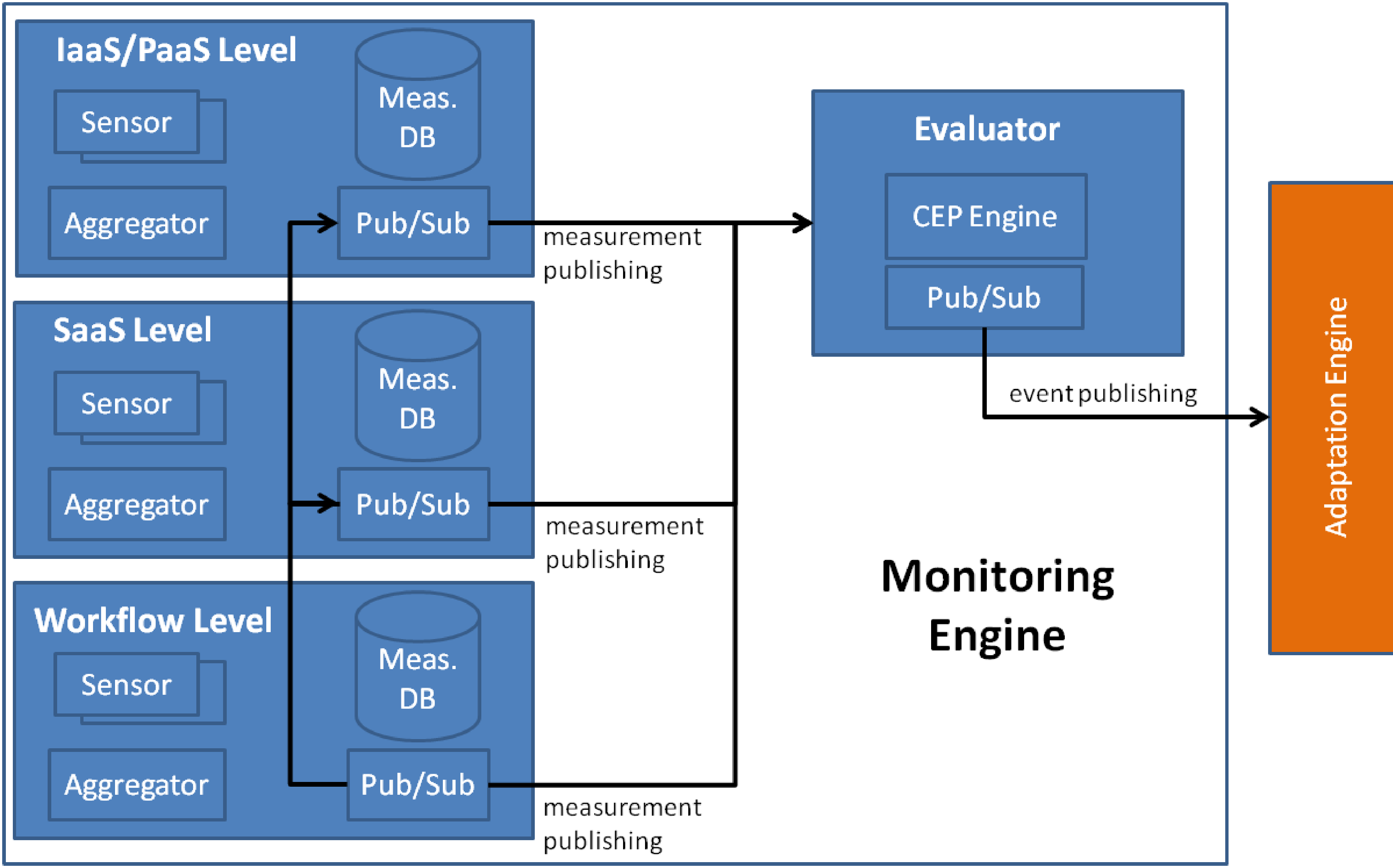
Solution Overview

- Overall BPaaS Management: CloudSocket project
  - Model-based approach for business-to-IT alignment & BPaaS provisioning
  - Lifecycle activity-specific environments
- BPaaS Monitoring:
  - Flexible metric specification via CAMEL [1]
  - Distributed monitoring approach across layers & clouds
    - CAMEL metric computation formulas/trees cover the measurability gap across layers & clouds
  - Layer-specific frameworks from FORTH [2] & UULM [3] Partners integrated to cover all layers
  - SLO-based evaluation mechanism [4] based on Complex Event Processing Engine

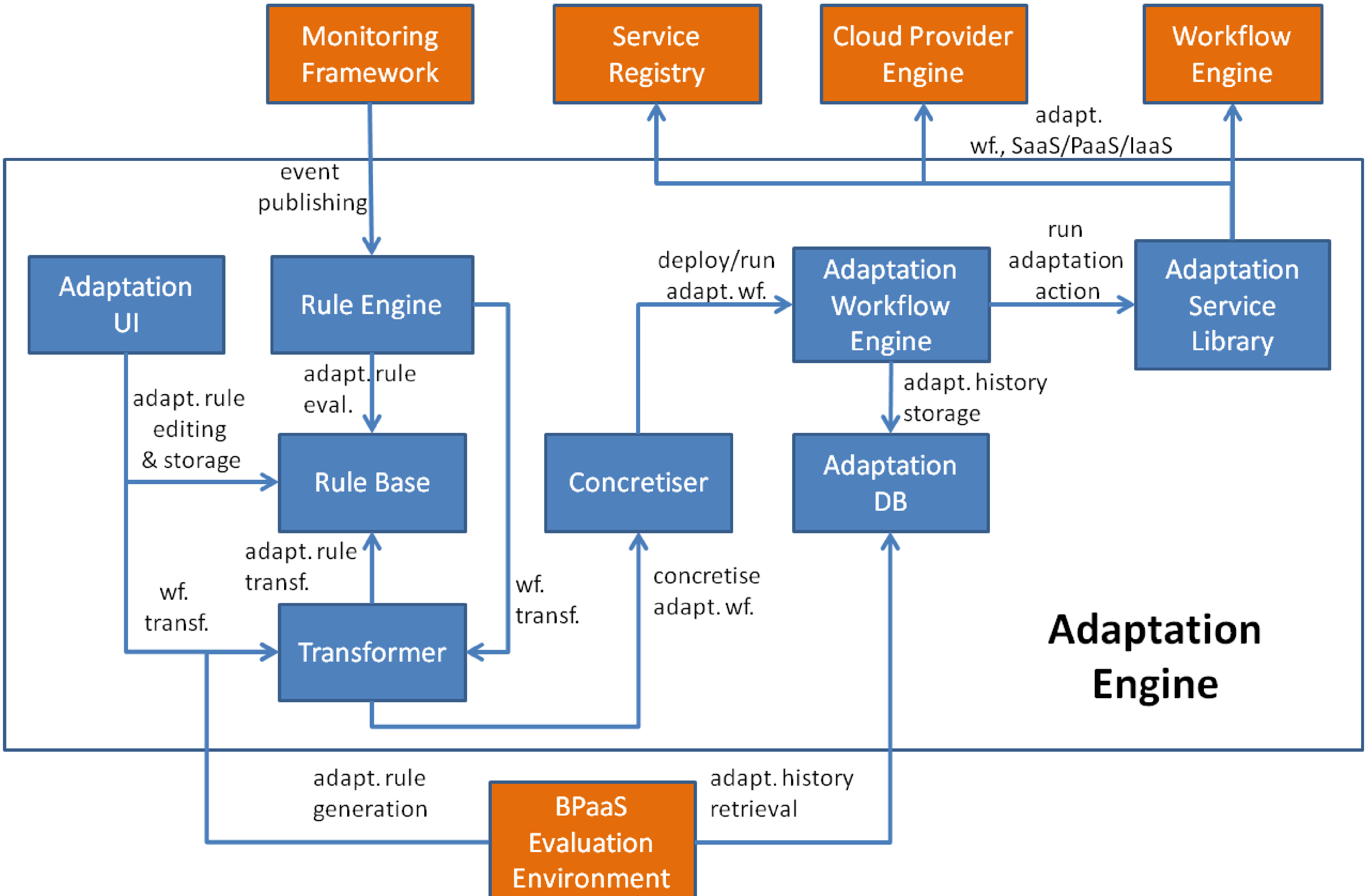
Solution Overview

- BPaaS Adaptation:
  - Composition of existing adaptation frameworks [2, 3]
    - To cover all possible layers
  - Pro- & re-active adaptation [2] via:
    - rule-based approach
    - event correlation via execution history mining
    - warning events
    - semi-automatic production of adaptation rules
  - Dynamic adaptation via concretisation & execution of abstract adaptation workflows specified in CAMEL
  - On-the-fly execution of adaptation workflows
  - Adaptation rule editing
  - Adaptation history recording & browsing

Solution Architecture



Solution Architecture



CAMEL – Overview

- Multi-DSL focusing on capturing different domain-specific aspects of multi-cloud applications:
  - Deployment, Requirement, Provider, Organisation, Location, Security, Metric, Scalability, Value Type, Unit
- Produced from existing languages (e.g., CloudML [5], Saloon [6]) & new ones (SRL [7])
- Use of OCL rules for integration & semantic domain validation

CAMEL – Overview

- Based on Eclipse EMF
  - Default tree-based editor
  - Programmatic support
- Text-based editor for devops based on XText technology
- More details:
  - www.camel-dsl.org
  - www.github.com/camel-dsl:
    - Meta-model
    - Domain-code
    - Text-based editor code

CloudSocket

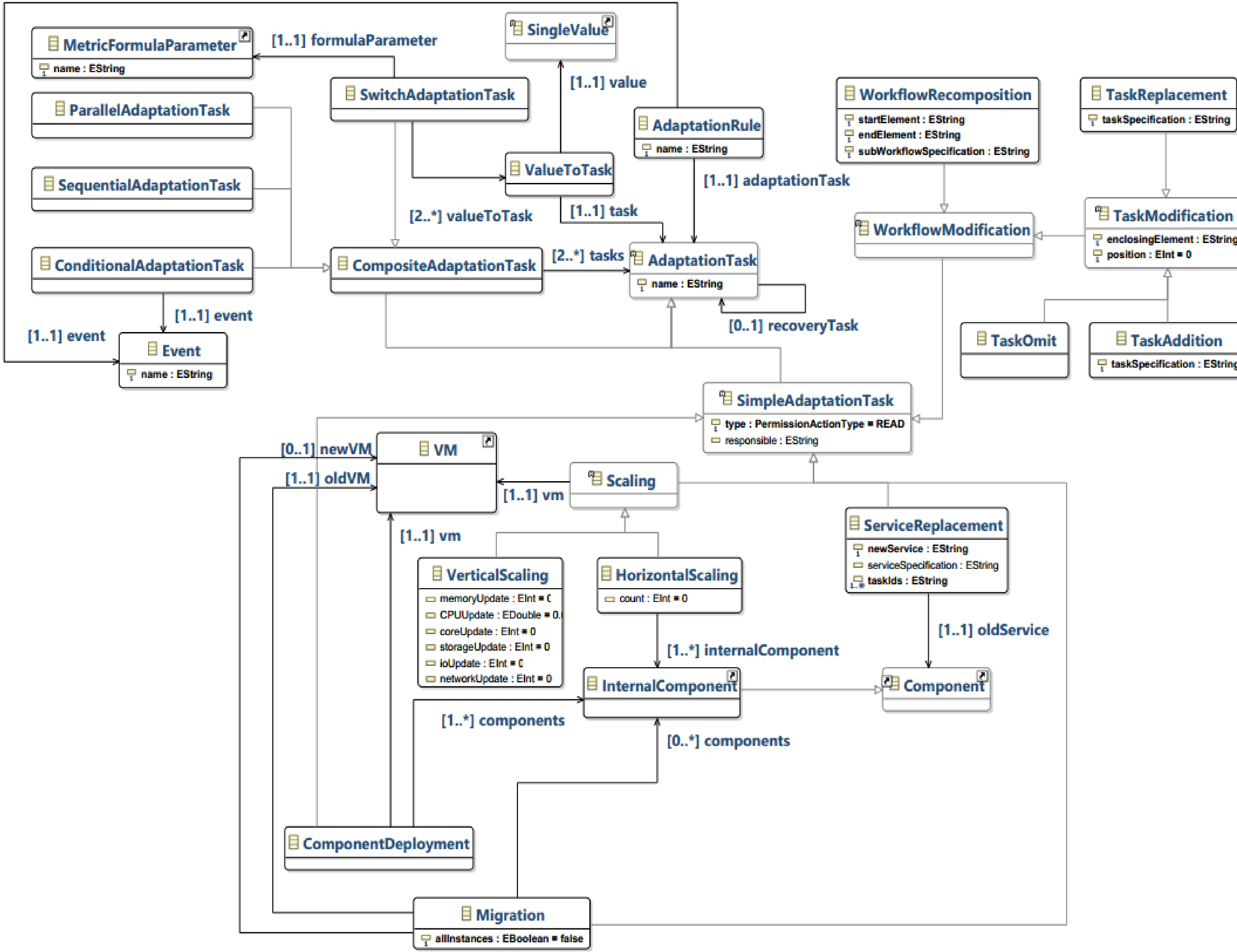
CAMEL – Overview

- Monitoring:
  - Specification of metric (computation) trees
  - Metric conditions
  - Metric scheduling & measurement window
- Scalability
  - Scalability rules mapping events to scaling actions
  - Both horizontal & vertical scaling actions supported
  - Events can be simple or composite
    - Simple events map to metric conditions
    - Composite events to event composition via temporal or logical operators

CAMEL – Extension

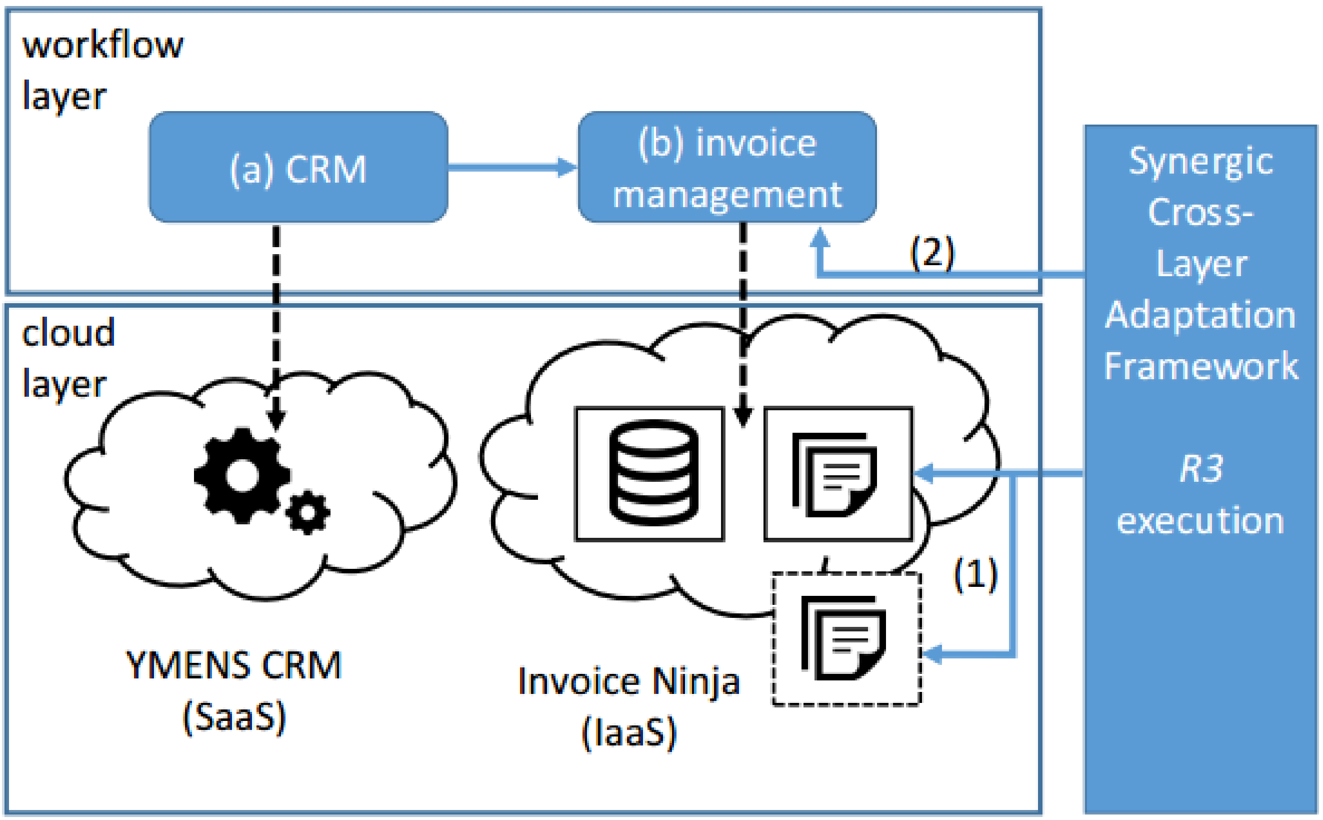
- Capability to specify complex adaptation actions instead of just scalability rules
  - Simple actions (*SimpleAdaptationTask*) mapping to layer-specific adaptation capabilities
    - Scale-in/out, Scale-up/down, Migration, Service Replacement, Workflow Recomposition, Task Add/Modify/Replace/Omit
  - Composite actions (CompositeAdaptationTask) mapping to a combination of actions via well-known control-flow constructs
    - Sequence, Parallel, Conditional, Switch
- Complex adaptation behaviour specified abstractly
  - Freedom to choose from alternative implementations of layer-specific simple adaptation actions

CloudSocket


MetricFormulaParameter
name : EString
[1..1] formulaParameter
SingleValue
[1..1] value
SwitchAdaptationTask
ParallelAdaptationTask
AdaptationRule
name : EString
WorkflowRecomposition
startElement : EString
endElement : EString
subWorkflowSpecification : EString
TaskReplacement
taskSpecification : EString
ValueToTask
SequentialAdaptationTask
[1..1] adaptationTask
[2..*] valueToTask
[1..1] task
TaskModification
enclosingElement : EString
position : EInt = 0
WorkflowModification
ConditionalAdaptationTask
CompositeAdaptationTask
[2..*] tasks
AdaptationTask
name : EString
[1..1] event
[0..1] recoveryTask
[1..1] event
Event
name : EString
TaskOmit
TaskAddition
taskSpecification : EString
SimpleAdaptationTask
type : PermissionActionType = READ
responsible : EString
[0..1] newVM
VM
[1..1] oldVM
Scaling
[1..1] vm
[1..1] vm
ServiceReplacement
newService : EString
serviceSpecification : EString
taskIds : EString
VerticalScaling
memoryUpdate : EInt = 0
CPUUpdate : EDouble = 0.
coreUpdate : EInt = 0
storageUpdate : EInt = 0
ioUpdate : EInt = 0
networkUpdate : EInt = 0
HorizontalScaling
count : EInt = 0
[1..1] oldService
[1..*] internalComponent
InternalComponent
Component
[1..*] components
[0..*] components
ComponentDeployment
Migration
allInstances : EBoolean = false

Synergic Cross-Layer Adaptation Framework Validation

- Cross-Layer Adaptation Scenario



Validation

- Initial Rule Set:
  - R1: cpu_viol(i_ninja,send_invoice) → hscale(i-ninja)
  - R2: down(i_ninja,send_invoice) → re-run (i-ninja)
- R2 covers non-permanent failures
- New rule is introduced by expert via CAMEL to handle permanent failures
  - R3: down(i_ninja,send_invoice) ∧ failed(R2) → seq(migrate(i_ninja), s_replace(i_ninja,send_invoice))

CloudSocket

Future Work

- Implementation and validation of distributed physical architectures for both frameworks
- Devise of sophisticated adaptation workflow concretisation algorithm
- Development of:
  - New adaptation capabilities
  - Alternative implementations of existing ones
- Dynamic injection of developed adaptation capabilities
- Adaptation history analysis
  - Statistical knowledge about successibility of rules and actions
  - Adjustment of adaptation rules
    - Update of semi-automatic adaptation rule derivation algorithm to exploit this knowledge

References

1. A. Rossini, K. Kritikos, N. Nikolov, J. Domaschka, F. Griesinger, D. Seybold, D. Romero, D2.1.3 – CloudML Implementation Documentation (Final version), PaaSage project deliverable, 2015.
2. Zeginis, C., Kritikos, K., Plexousakis, D.: Event pattern discovery in multi-cloud service-based applications. IJSSOE 5(4), 78–103 (2015)
3. Domaschka, J., Seybold, D., Griesinger, F., Baur, D.: Axe: A novel approach for generic, flexible, and comprehensive monitoring and adaptation of cross-cloud applications. In: European Conference on Service-Oriented and Cloud Computing, pp. 184–196. Springer (2015)
4. Kritikos, K., Zeginis, C., Paravoliasis, A., Plexousakis, D.: CEP-Based SLO Evaluation. In: BPM@Cloud Workshop in ESOCC. Springer (2017)
5. N. Ferry, A. Rossini, F. Chauvel, B. Morin, A. Solberg, Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems, in: L. O'Conner (Ed.), Proceedings of CLOUD 2013: 6*th* IEEE International Conference on Cloud Computing, IEEE Computer Society, ISBN 978-0-7695-5028-2, 887–894, 2013.
6. C. Quinton, D. Romero, L. Duchien, Cardinality-based feature models with constraints: a pragmatic approach, in: T. Kishi, S. Jarzabek, S. Gnesi (Eds.), SPLC 2013: 17th International Software Product
Line Conference, ACM, 162–166, 2013
7. K. Kritikos, J. Domaschka, A. Rossini, SRL: A Scalability Rule Language for Multi-cloud Environments, in: CloudCom, IEEE, 1–9, 2014.